User Manual

32

eFLASHLOAD

embedded FLASH Programmer Utility for V850 and RH850 based 32-bit Microcontrollers

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

R01UT0177ED0301, Rev. 3.01
January 29, 2015

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

8. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti- crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems;medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

9. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

10. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

11. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

13. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

    (Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

    (Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Regional Information

Some information contained in this document may vary from country to country. Before using any Renesas Electronics product in your application, please contact the Renesas Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

Visit

http://www.renesas.com

to get in contact with your regional representatives and distributors.

Table of Contents

Chapter 1 Warranty Disclaimer 8
1.1 Limitation of Liability 8

Chapter 2 Introduction 9

Chapter 3 FLASH monitor program 11
3.1 Parameter / Command Interface 11
3.2 FLASH Monitor Commands 13
3.2.1 Blank check command 14
3.2.2 Erase command 15
3.2.3 Program command 16
3.2.4 Verify command 17
3.2.5 User Command 19
3.3 FLASH monitor program structure 20
3.4 FLASH monitor memory mapping 23

Chapter 4 eFLASHLOAD GUI 25
4.1 eFLASHLOAD Menu bar 26
4.1.1 [File] menu 26
4.1.2 [Setup] menu 29
4.1.3 [Help] menu 30
4.2 eFLASHLOAD Tool bar 31
4.2.1 Device Setup dialog 32
4.2.2 FLASH Monitor Setup dialog 34
4.2.3 FLASH Image setup dialog 36
4.2.4 Connect / Disconnect button 38
4.2.5 Blank check button 38
4.2.6 Erase button 38
4.2.7 Program button 38
4.2.8 Verify button 38
4.2.9 E.P.V. button 38
4.2.10 User button 38
4.2.11 Delete button 38
4.3 Action log window 39
4.4 Programming Parameter Window 40
4.5 eFLASHLOAD Status bar 41

Chapter 5 Command line interface 42
5.1 Command line options 43
5.2 Return Codes 44

List of Figures

Figure 1 eFLASHLOAD System Overview....10
Figure 2 Command / Parameter area....11
Figure 3 Example of verify function....18
Figure 4 FLASH monitor program structure....20
Figure 5 FLASH monitor prototypes and defines....22
Figure 6 FLASH monitor memory mapping....24
Figure 7 Graphical user interface....25
Figure 8 File menu....26
Figure 9 XML based configuration file....27
Figure 10 Setup menu, auto connect off....29
Figure 11 Setup menu, auto connect on....29
Figure 12 About eFLASHLOAD dialog....30
Figure 13 eFLASHLOAD tool bar....31
Figure 14 Device Setup Dialog....32
Figure 15 Device and Config file information....33
Figure 16 FLASH Monitor Setup dialog....34
Figure 17 FLASH Monitor File information....35
Figure 18 FLASH Image Setup dialog....36
Figure 19 FLASH Image File information....37
Figure 20 Action log window....39
Figure 21 Programming Parameter Window....40
Figure 22 Status bar....41
Figure 23 Command line interface....42
Figure 24 Command line error message....43

List of Tables

Table 1 eFL structure elements 13
Table 2 Monitor commands 13
Table 3 Blank check command 14
Table 4 Erase command 15
Table 5 Program command 16
Table 6 Verify command 17
Table 7 User command 19
Table 8 Main window areas 26
Table 9 XML predefined tags 28
Table 10 Status bar 41
Table 11 Command line options 43

Chapter 1 Warranty Disclaimer

Because eFLASHLOAD and the FLASH Monitor examples are licensed free of charge, there is no warranty of any kind whatsoever and expressly disclaimed and excluded by Renesas, either expressed or implied, including but not limited to those for non-infringement of intellectual property, merchantability and/or fitness for the particular purpose. Renesas shall not have any obligation to maintain, service or provide bug fixes for the supplied Product(s) and/or the Application.

Each User is solely responsible for determining the appropriateness of using eFLASHLOAD and the FLASH Monitor examples and assumes all risks associated with its exercise of rights under this Agreement, including, but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

1.1 Limitation of Liability

In no event shall Renesas be liable to the User for any incidental, consequential, indirect, or punitive damage (including but not limited to lost profits) regardless of whether such liability is based on breach of contract, tort, strict liability, breach of warranties, failure of essential purpose or otherwise and even if advised of the possibility of such damages. Renesas shall not be liable for any services or products provided by third party vendors, developers or consultants identified or referred to the User by Renesas in connection with eFLASHLOAD and the FLASH Monitor examples.

Chapter 2 Introduction

eFLASHLOAD is a software which allows the programming of embedded FLASH memories connected to a V850 or RH850 based 32-bit Microcontroller. The eFLASHLOAD GUI uses the standard Renesas Debug hardware for programming, as they were the QB-V850MINI-L On-Chip Debug Emulator, the E1 On-Chip Debug Emulator, the IECUBE and IECUBE2 In-Circuit-Emulators. eFLASHLOAD uses the standard debug interface to communicate with the target microcontroller. For V850ES and V850E1 based devices the NWIRE debug interface is used for communication, respectively for V850E2 the NEXUS debug interface. Both debugs interface types, NWIRE as well NEXUS, are JTAG based communication interfaces. For RH850 based microcontrollers the Renesas proprietary Low-Pin-Debug interface, shortened LPD, is used. The LPD interface supports a 1-pin as well 4-pin debug communication to a RH850 based target microcontroller.

The software supports typical FLASH programmer functions as Blank check, Erase, Program, Verify and the sequential E.P.V. operation (Erase -> Program -> Verify). The software consists of a Windows GUI which controls a FLASH monitor program running on the target microcontroller. The communication and synchronisation between the GUI and the FLASH monitor program is done via a standardized FLASH Loader parameter and command interface. The FLASH monitor program running on the V850 or RH850 based target microcontroller is under full user responsibility and control. Therefore there is no restriction on the supported FLASH memories.

By starting the eFLASHLOAD software the FLASH monitor program is loaded to the internal RAM of the V850 or RH850 based target microcontroller and executed. All programming functions can be controlled by the eFLASHLOAD GUI in the same manner as using a standard FLASH programmer GUI like PG-FP5.

The minimum system requirements for using eFLASHLOAD were:

- Target hardware consisting of
  - V850 or RH850 based microcontroller
  - embedded FLASH memory
  - On-Chip debug interface or emulator interface

- V850 or RH850 based target microcontroller
  - iRAM resources:
    - 4124 Bytes for the FLASH Monitor Parameter / Command area
    - 4 kByte ~ 16 kByte for the FLASH Monitor Program (depending on user implementation)
  - Debug interfaces:
    - NWIRE (JTAG based) → V850ES, V850E1
    - NEXUS (JTAG based) → V850E2
    - 1-Pin and 4-Pin LPD (Low Pin Debug) → RH850

- Debug Hardware
  - QB-V850MINI-L On-Chip debug emulator
  - E1 On-Chip debug emulator
  - IECUBE and IECUBE2 In-Circuit-Emulators

- Windows XP, Windows Vista or Windows 7 based PC / Workstation

Note: By using the In-Circuit-Emulators IECUBE or IECUBE2, the emulator has to be supplied with target power.

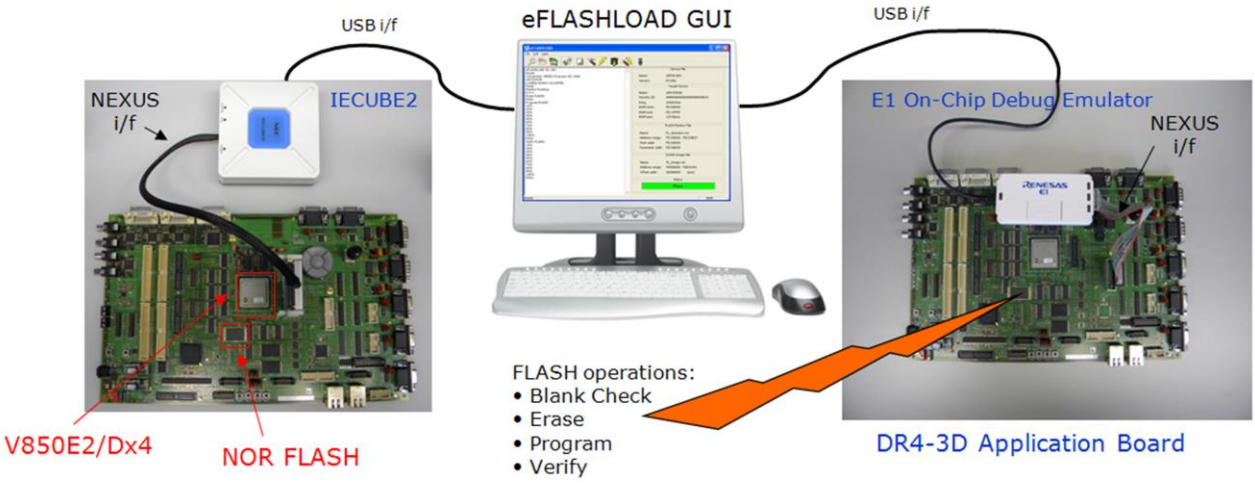


Figure 1 eFLASHLOAD System Overview

Chapter 3 FLASH monitor program

The eFLASHLOAD software installation does include a template of a FLASH monitor program. All FLASH vendor specific functions as FLASH initialisation, blank check, erase, program or verify have to be implemented by the user itself.

3.1 Parameter / Command Interface

The FLASH monitor program operation is based and controlled on a common parameter / command interface. The parameter / command interface uses a reserved memory area within the internal RAM of the microcontroller. The following structure is placed into the FLASH monitor parameter area which controls the FLASH monitor operation and also indicates the status and result of a FLASH operation. Additionally a 4kByte transfer buffer is included within the structure, which stores the data to be programmed or verified during FLASH operation.

```
// FLASH Monitor Parameter Area
#pragma ghs startdata
#pragma ghs section bss=".eFLASHLoad"
volatile struct {
    unsigned int  H_cmd;        // (HOST   -> TARGET) command
    unsigned int  T_cmd_ack;    // (TARGET -> HOST  ) command acknowledgement
    unsigned int  T_cmd_stat;   // (TARGET -> HOST  ) command status
    unsigned int  T_cmd_info;   // (TARGET -> HOST  ) command information
    unsigned int  H_fps_addr;   // (HOST   -> TARGET) FLASH programming start address
    unsigned int  H_fpe_addr;   // (HOST   -> TARGET) FLASH programming end address
    unsigned int  H_buf_size;   // (HOST   -> TARGET) buffer size, amount of bytes within buffer
    unsigned int  H_buf[1024];  // (HOST   -> TARGET) buffer, 4kByte transfer buffer
    }eFL;
#pragma ghs section bss=default
#pragma ghs enddata
```

Figure 2 Command / Parameter area

The following table describes the elements of the eFL structure.

Type	Argument	Description
unsigned integer	eFL.H_cmd	FLASH monitor command to be executed. The variable is written by eFLASHLOAD (Host) and controls the operation of the FLASH monitor program (target). The following commands are supported: **CMD_BLANK** **CMD_ERASE** **CMD_PROGRAM** **CMD_VERIFY** **CMD_USER** The FLASH monitor program reads this variable and starts the indicated command execution.

unsigned integer	eFL.T_cmd_ack	The FLASH monitor program (target) writes to this variable to acknowledge a command indicated by eFLASHLOAD (host). The target has to acknowledge the same command code indicated by the host. The following acknowledge commands are supported: **CMD_BLANK** **CMD_ERASE** **CMD_PROGRAM** **CMD_VERIFY** **CMD_USER** A cyclic read operation is performed by the eFLASHLOAD GUI to get the operating status. In case a wrong command is acknowledge by the FLASH monitor program or the command is not acknowledged in time, the operation is stopped by the eFLASHLOAD GUI and an error is displayed.
unsigned integer	eFL.T_cmd_stat	The FLASH monitor program (target) notifies the status or results of the current command under execution. The following states are supported: **BUSY** **PASS** **FAIL** A cyclic read operation is performed by the eFLASHLOAD GUI to get the operating status and result.
unsigned integer	eFL.T_cmd_info	The command info variable shows the current FLASH access address during FLASH programming and verifying. This variable has to be updated by the FLASH monitor program within the program and verify function / routine. The eFLASHLOAD GUI (host) does use this information to show and update the progress status during programming and verification. A cyclic read operation is performed by the host to update command progress information.
unsigned integer	eFL.H_fps_addr	This variable is written by eFLASHLOAD (host) and defines the FLASH start address for the current command.
unsigned integer	eFL.H_fpe_addr	This variable is written by eFLASHLOAD (host) and defines the FLASH end address for the current command.
unsigned integer	eFL.H_buf_size	This variable is written by eFLASHLOAD (host) and defines the amount of bytes within the transfer buffer. The maximum buffer size is equal to 4096 Bytes.

unsigned integer	eFL.H_buf[1024]	This is the transfer buffer for data to be written to the FLASH memory or data to be verified from the FLASH memory. The transfer buffer is written by eFLASHLOAD. The buffer size is fixed to 1024 words (4096 bytes).

Table 1 eFL structure elements

3.2 FLASH Monitor Commands

The following commands are supported by eFLASHLOAD. By starting a FLASH operation using the eFLASHLOAD GUI the corresponding command code and FLASH parameter set is written to the V850 or RH850 based target microcontroller. The FLASH monitor program has to acknowledge the received command in time. After this the selected operation is started. The eFLASHLOAD software performs a cyclic reading of the command status register to check the current command status.

Command	Code	Value
Blank check	CMD_BLANK	0x11111111
Erase	CMD_ERASE	0x22222222
Program	CMD_PROGRAM	0x33333333
Verify	CMD_VERIFY	0x44444444
User	CMD_USER	0x77777777

Table 2 Monitor commands

3.2.1 Blank check command

Blank Check Command: **CMD_BLANK** (= 0x11111111)

FLASH Monitor Function: ***unsigned int FLM_Blank()***

Input Parameters:

Parameter	Input Value	Remark
eFL.H_fps_addr	FLASH start address	
eFL.H_fps_end	FLASH end address	

Return Parameters:

Parameter	Return Value	Remark
unsigned int	PASS, FAIL	
eFL.H_fps_end	In case of BLANK check failure, the error address is returned.	

Table 3 Blank check command

3.2.2 Erase command

Erase Check Command: **CMD_ERASE** (= 0x22222222)

FLASH Monitor Function: ***unsigned int FLM_Erase()***

Input Parameters:

Parameter	Input Value	Remark
eFL.H_fps_addr	FLASH start address	
eFL.H_fps_end	FLASH end address	

Return Parameters:

Parameter	Return Value	Remark
unsigned int	PASS, FAIL	
eFL.H_fps_end	In case of ERASE failure, the error address is returned.	

Table 4 Erase command

3.2.3 Program command

Program Command: **CMD_PROGRAM** (= 0x33333333)

FLASH Monitor Function: ***unsigned int FLM_Program()***

Input Parameters:

Parameter	Input Value	Remark
eFL.H_fps_addr	FLASH operation start address	
eFL.H_buf_size	Amount of bytes to be programmed	
*eFL.H_buf	Data to be written to the FLASH memory	

Update Parameters:

Parameter	Value	Remark
eFL.T_cmd_info	Current FLASH access address during programming	This variable has to be updated in the FLASH programming function, to indicate the progress status.

Return Parameters:

Parameter	Return Value	Remark
unsigned int	PASS, FAIL	
eFL.H_fps_end	In case of PROGRAM failure, the error address is returned.	

Table 5 Program command

3.2.4 Verify command

Verify Command: **CMD_VERIFY** (= 0x44444444)

FLASH Monitor Function: ***unsigned int FLM_Verify()***

Input Parameters:

Parameter	Input Value	Remark
eFL.H_fps_addr	FLASH operation start address	
eFL.H_buf_size	Amount of bytes to be verified	
*eFL.H_buf	Data to be verified from the FLASH memory	

Update Parameters:

Parameter	Value	Remark
eFL.T_cmd_info	Current FLASH access address during verifying	This variable has to be updated in the FLASH verify function, to indicate the progress status.

Return Parameters:

Parameter	Return Value	Remark
unsigned int	PASS, FAIL	
eFL.H_fps_end	In case of VERIFY failure, the error address is returned.	
eFL.H_fpe_end	In case of VERIFY failure, the error data is returned.	

Table 6 Verify command

An example of the FLASH monitor verification function could be similar to the following implementation.

```
//=========================================================================
//  Function:       FLM_Verify
//  Description :   Executes verify command
//=========================================================================
unsigned int FLM_Verify(void)
{
    // pointer to transfer buffer, reference data
    unsigned int *verify_addr;
    unsigned int verify_data;

    // pointer to FLASH memory, verification data
    unsigned int *flash_read_addr;
    unsigned int  flash_read_data;

    // get FLASH start address, verification data start address
    flash_read_addr = (unsigned int *) eFL.H_fps_addr;
    // get transfer buffer start address, the reference base
    verify_addr     = (unsigned int) &eFL.H_buf;

    // start verify
    while(flash_read_addr <= (unsigned int *) (eFL.H_fps_addr + eFL.H_buf_size))
    {
        // get FLASH contents, verification data
        flash_read_data  = *flash_read_addr;
        // get transfer buffer, reference data
        verify_data      = *verify_addr;

        // indicate current FLASH address, for progress information displayed
        // by the eFLASHLOAD GUI
        eFL.T_cmd_info = (unsigned int) flash_read_addr;

        // compare FLASH data with reference data
        if(flash_read_data != verify_data)
        {
            // in case of mismatch, indicate FLASH address and FLASH read data
            // to be displayed by the eFLASHLOAD GUI
            eFL.H_fps_addr  = (unsigned int) flash_read_addr;
            eFL.H_fpe_addr  = flash_read_data;
            // return verify failure
            return FAIL;
        }
        // increment address pointers
        *flash_read_addr++;
        *verify_addr++;
    }
    // return verify success
    return PASS;
}
```

Figure 3 Example of verify function

3.2.5 User Command

User Command: **CMD_USER** (= 0x77777777)

FLASH Monitor Function: ***unsigned int FLM_User()***

Input Parameters:

None

Return Parameters:

Parameter	Return Value	Remark
unsigned int	PASS, FAIL	
eFL.H_fps_end	A user specific value can be returned.	

Table 7 User command

The User Command is a freely configurable function supported by the eFLASHLOAD GUI and the FLASH monitor program. It can be used to execute for instance test software or any other user code before or after a dedicated FLASH operation. The corresponding user code has simply to be added to the FLASH monitor program.

3.3 FLASH monitor program structure

As reference a skeleton of the FLASH monitor program running on the V850 target microcontroller is shown in the figure below.

```
void main (void)
{
// Initialisation of CPU
// TODO start: add code here

// TODO end

    if(FLM_Init()==FAIL)     eFL.T_cmd_stat = FAIL;
    else                     eFL.T_cmd_stat = PASS;

    while(1)
    {
        switch(eFL.H_cmd)
        {
            case CMD_BLANK:      eFL.T_cmd_ack  = CMD_BLANK;
                                 eFL.T_cmd_stat = BUSY;

                                 if(FLM_Blank()==FAIL)   eFL.T_cmd_stat = FAIL;
                                 else                    eFL.T_cmd_stat = PASS;
                                 eFL.H_cmd = PASS;
                                 break;

            case CMD_ERASE:      eFL.T_cmd_ack  = CMD_ERASE;
                                 eFL.T_cmd_stat  = BUSY;

                                 if(FLM_Erase()==FAIL)   eFL.T_cmd_stat = FAIL;
                                 else                    eFL.T_cmd_stat = PASS;
                                 eFL.H_cmd = PASS;
                                 break;

            case CMD_PROGRAM:    eFL.T_cmd_ack  = CMD_PROGRAM;
                                 eFL.T_cmd_stat = BUSY;

                                 if(FLM_Write()==FAIL)   eFL.T_cmd_stat = FAIL;
                                 else                    eFL.T_cmd_stat = PASS;
                                 eFL.H_cmd = PASS;
                                 break;

            case CMD_VERIFY:     eFL.T_cmd_ack  = CMD_VERIFY;
                                 eFL.T_cmd_stat = BUSY;

                                 if(FLM_Verify()==FAIL)  eFL.T_cmd_stat = FAIL;
                                 else                    eFL.T_cmd_stat = PASS;
                                 eFL.H_cmd  = PASS;
                                 break;

            case CMD_USER:       eFL.T_cmd_ack  = CMD_USER;
                                 eFL.T_cmd_stat = BUSY;

                                 if(FLM_User()==FAIL)    eFL.T_cmd_stat = FAIL;
                                 else                    eFL.T_cmd_stat = PASS;
                                 eFL.H_cmd  = PASS;
                                 break;

            default:             break;
        }
    }
}
```

Figure 4 FLASH monitor program structure

After entering the main function, the FLASH is initialized by calling function “FLM_Init()”. The init function returns a PASS or FAIL information, which is checked by eFLASHLOAD. Within the endless loop the main body of the FLASH monitor program can be found. The monitor program reads the FLASH monitor command variable “eFL.H_cmd” written by eFLASHLOAD and branches to the corresponding case.

Within the case the command is acknowledge by the monitor by writing to the FLASH monitor command acknowledge register “eFL.T_cmd_ack”. After this the command status register is changed to BUSY and the command is executed by calling the corresponding FLASH function as they are FLM_Blank(), FLM_Erase(), FLM_Write(), FLM_Verify() and FLM_User().

The corresponding FLASH function returns a PASS or FAIL information, which indicates if the FLASH operation was successfully. The result is checked by eFLASHLOAD. After a FLASH operation is finished the FLASH monitor command variable is set to PASS and the FLASH monitor program is ready to receive the next command from the host (eFLASHLOAD).

The following prototypes and defines are used by the FLASH monitor program.

```
// FLASH Monitor Functions
unsigned int FLM_Init(void);
unsigned int FLM_Blank(void);
unsigned int FLM_Erase(void);
unsigned int FLM_Write(void);
unsigned int FLM_Verify(void);
unsigned int FLM_User(void);

// FLASH Monitor Defines
#define CMD_BLANK   0x11111111
#define CMD_ERASE   0x22222222
#define CMD_PROGRAM 0x33333333
#define CMD_VERIFY  0x44444444
#define CMD_USER    0x77777777

#define BUSY        0x55555555
#define PASS        0x00000000
#define FAIL        0xffffffff
```

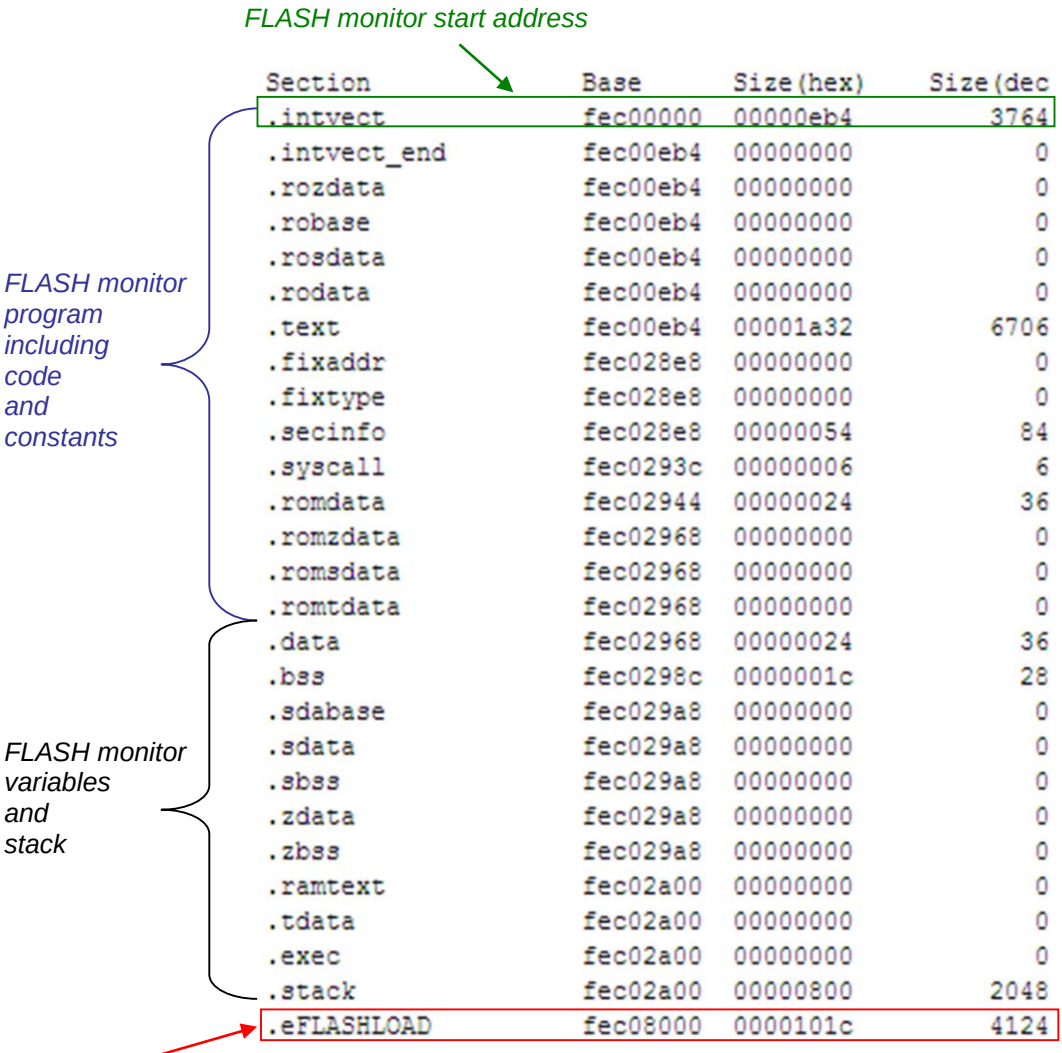
Figure 5 FLASH monitor prototypes and defines

3.4 FLASH monitor memory mapping

The FLASH programming GUI allows and restricts the monitor program download and monitor execution only to and from the internal RAM of the target microcontroller. Downloading to the internal FLASH memory of the microcontroller is not supported. The eFLASHLOAD GUI checks the mapping of the FLASH monitor program code which must fit to the iRAM area of the V850 or RH850 based microcontroller.

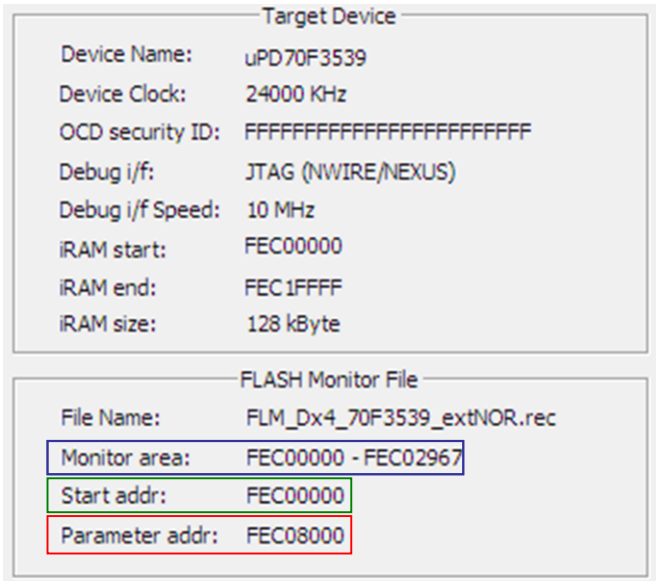
Additionally to this, the FLASH parameter / area which is used as communication and synchronisation interface between the eFLASHLOAD GUI and the FLASH monitor program must be placed to the internal RAM of the microcontroller as well.

A sample of the FLASH monitor memory mapping for a V850E2/Dx4 device and the Green Hills tool chain is shown in the next figure.


FLASH monitor start address
FLASH monitor program including code and constants
FLASH monitor variables and stack
FLASH monitor parameter area (user defined section)

Section	Base	Size(hex)	Size(dec
.intvect	fec00000	00000eb4	3764
.intvect_end	fec00eb4	00000000	0
.rozdata	fec00eb4	00000000	0
.robase	fec00eb4	00000000	0
.rosdata	fec00eb4	00000000	0
.rodata	fec00eb4	00000000	0
.text	fec00eb4	00001a32	6706
.fixaddr	fec028e8	00000000	0
.fixtype	fec028e8	00000000	0
.secinfo	fec028e8	00000054	84
.syscall	fec0293c	00000006	6
.romdata	fec02944	00000024	36
.romzdata	fec02968	00000000	0
.romsdata	fec02968	00000000	0
.romtdata	fec02968	00000000	0
.data	fec02968	00000024	36
.bss	fec0298c	0000001c	28
.sdabase	fec029a8	00000000	0
.sdata	fec029a8	00000000	0
.sbss	fec029a8	00000000	0
.zdata	fec029a8	00000000	0
.zbss	fec029a8	00000000	0
.ramtext	fec02a00	00000000	0
.tdata	fec02a00	00000000	0
.exec	fec02a00	00000000	0
.stack	fec02a00	00000800	2048
.eFLASHLOAD	fec08000	0000101c	4124

FLASH monitor memory mapping



Corresponding eFLASHLOAD setting and configuration

Figure 6 FLASH monitor memory mapping

Chapter 4 eFLASHLOAD GUI

The graphical user interface of eFLASHLOAD allows an easy and an intuitive configuration and handling. When starting the eFLASHLOAD GUI a screen similar to the following one will appear. Some commands and functions me be unavailable when the GUI is started, depending on the specified configuration.

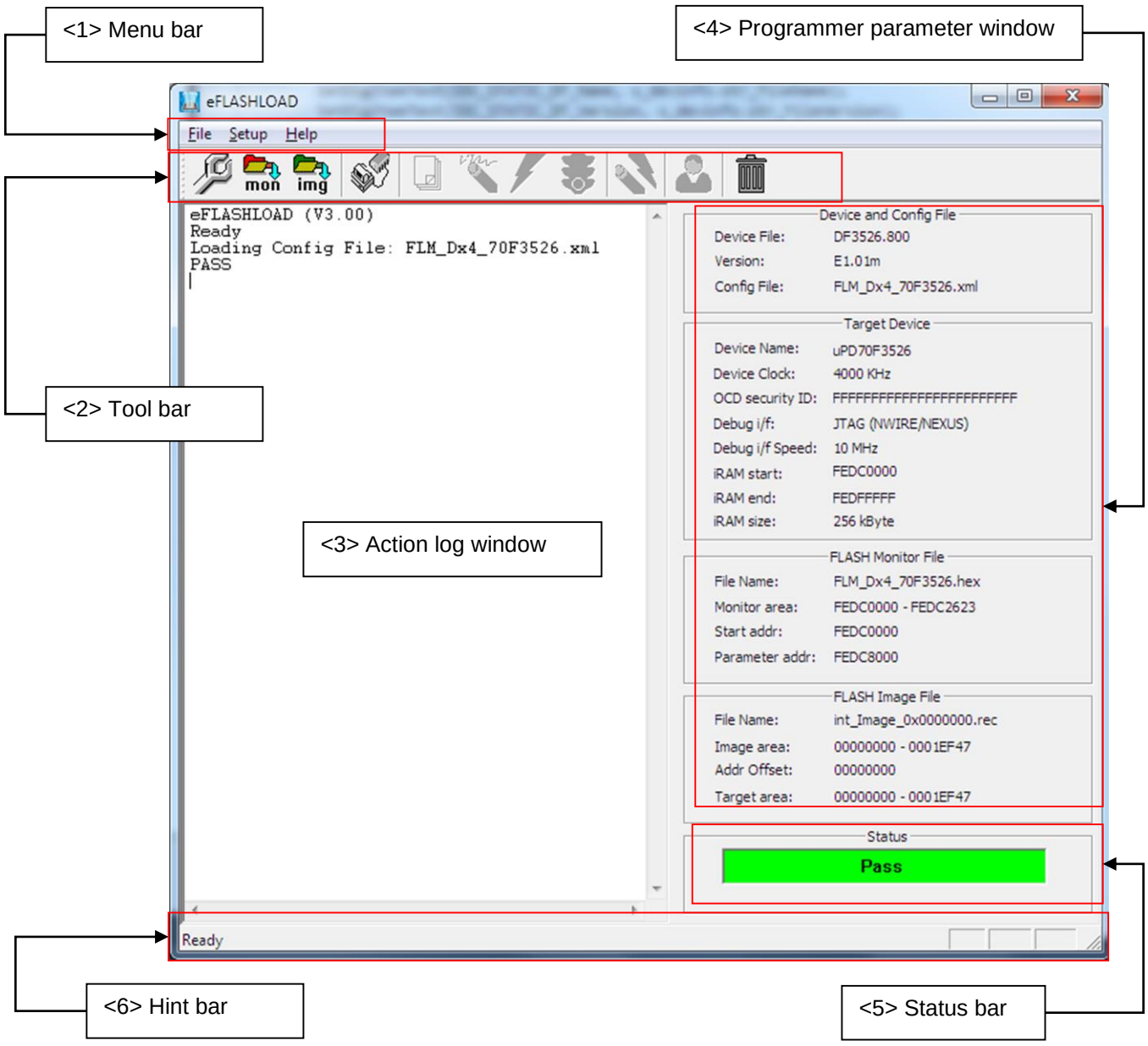


Figure 7 Graphical user interface

The main window consists of the following areas.

Name		Displayed Items
<1>	Menu bar	Menu items executable by the programming GUI
<2>	Tool bar	Frequently used commands, as buttons
<3>	Action log window	A programming GUI action log
<4>	Programmer parameter window	Programming parameter settings
<5>	Status bar	Command progress shown as a color or with a message
<6>	Hint bar	Hints for commands and toolbar

Table 8 Main window areas

4.1 eFLASHLOAD Menu bar

The menu bar of the embedded FLASH programmer GUI allows to load or save the tool configuration and to configure the GUI.

4.1.1 [File] menu

The file menu allows to load an existing or stored the current FLASH programmer configuration. The configuration consists of the target device, FLASH monitor and FLASH image settings. The configuration is stored within the XML file format.

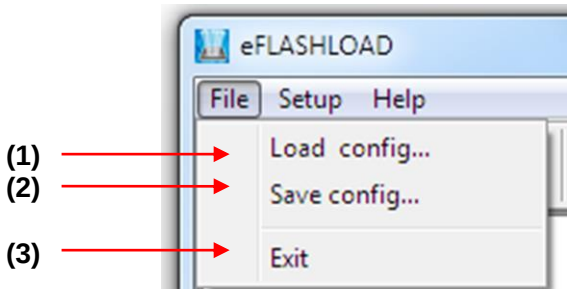


Figure 8 File menu

(1) Load XML based configuration file.

(2) Save XML based configuration file.

(3) Exit FLASH programmer GUI.

The XML based configuration file and the corresponding tags are shown below.

```
<?xml version="1.0"?>
<eFLASHLOAD>
   <DeviceSection>
      <DEVICE_FILE>..\devicefile\dr7f701035.dvf</DEVICE_FILE>
      <DEVICE_OCDSECID>FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</DEVICE_OCDSECID>
      <DEVICE_CLOCK>8000</DEVICE_CLOCK>
      <DEVICE_DEBUGINTERFACE>LPD4PIN</DEVICE_DEBUGINTERFACE>
      <DEVICE_DEBUGSPEED>11000</DEVICE_DEBUGSPEED>
      <DEVICE_AUTOCON>off</DEVICE_AUTOCON>
   </DeviceSection>
   <MonitorSection>
      <MONITOR_FILE>..\FLASH_monitor\FLM_F1L_R7F701035_Selfprog\out\FLM_F1L_R7F701035.hex</MONITOR_FILE>
      <MONITOR_SADDR>FEDE0000</MONITOR_SADDR>
      <MONITOR_PADDR>FEDE8000</MONITOR_PADDR>
   </MonitorSection>
   <FlashSection>
      <FLASH_FILE>..\_image\ext_Image_0xF0000000.hex</FLASH_FILE>
      <FLASH_ADDROFFSET>F0000000</FLASH_ADDROFFSET>
   </FlashSection>
</eFLASHLOAD>
```

Figure 9 XML based configuration file

The predefined tags are:

eFLASHLOAD	**Defining the start of the configuration file**
DeviceSection	Defining the device setup configuration
DEVICE_FILE	Specifies the device file
DEVICE_OCDSECID	Specifies the OCD security ID
DEVICE_CLOCK	Specifies the device clock (target frequency)
DEVICE_DEBUGINTERFACE	Specifies the debug interface type: • For V850 based products this tag is ignored. The debug interface is fixed to JTAG. • For RH850 based products the 1-pin LPD or 4-pin LPD Debug Interface must be chosen. The following settings are supported: **LPD1PIN** **LPD4PIN**
DEVICE_DEBUGSPEED	Specifies the speed of the debug interface: • For V850 based products this tag is ignored. The debug interface speed is fixed to 10MHz. • For RH850 based products please specify the speed as following: → For 1-pin LPD specify a baudrate of: **2000**, **1000** or **500** Kbps. → For 4-pin LPD specify a LPD clock of: **16500**, **11000** or **5500** KHz.
DEVICE_AUTOCON	Specifies if auto connection is enabled or disabled. Possible settings are: on / off
MonitorSection	Defining the FLASH monitor setup configuration
MONITOR_FILE	Specifies the FLASH monitor file
MONITOR_SADDR	Specifies the FLASH monitor start address
MONITOR_PADDR	Specifies the FLASH monitor parameter area base address
FlashSection	Defining the FLASH image setup configuration
FLASH_FILE	Specifies the FLASH image file
FLASH_ADDROFFSET	Specifies the FLASH address offset

Table 9 XML predefined tags

4.1.2 [Setup] menu

The setup menu allows the configuration of the connection mode. By selecting “Auto connect” to off, the user has to establish the debug communication to the target device manually. This is done by pressing the connect button within the tool bar. After the corresponding FLASH operation was executed, for instance blank check, erase, program or verify, the user has to close the debug communication by pressing the connect button ones again. Only if “Auto connect” is to off, the connect button is visible.

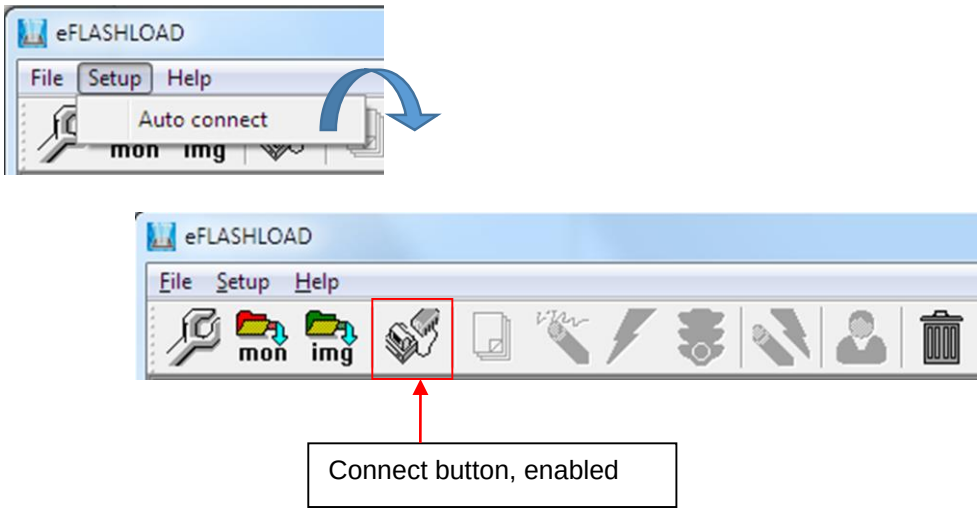


Figure 10 Setup menu, auto connect off

In case “Auto connect” is set to on, the communication to the target device is establish automatically. By performing a corresponding FLASH operation, for instance blank check, erase, program or verify, the debug communication is established first, the FLASH operation is executed and afterwards the debug communication is closed. The connect button is disabled in case “Auto connect” is set to on.

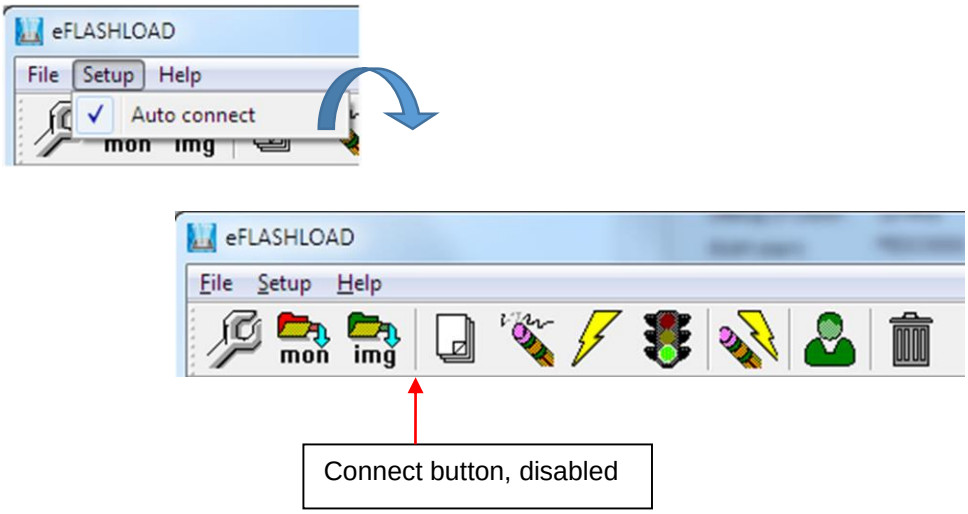


Figure 11 Setup menu, auto connect on

4.1.3 [Help] menu

The help menu shows the release and version information of the FLASH programmer GUI.

Figure 12 About eFLASHLOAD dialog

4.2 eFLASHLOAD Tool bar

The tool bar contains buttons to configure the embedded FLASH programmer GUI and to start all key functions quickly. The following buttons and corresponding functions are offered by eFLASHLOAD.

	Opens the Device Setup dialog box.
	Opens the FLASH Monitor Setup dialog box.
	Opens the FLASH Image Setup dialog box.
	Executes the connect command or disconnect command.
	Executes the [Blank Check] command.
	Executes the [Erase] command.
	Executes the [Program] command.
	Executes the [Verify] command.
	Executes the [Autoprocedure(E.P.V.)] command. (Erase -> Program -> Verify).
	Executes the [User] command.
	Deletes all settings

Figure 13 eFLASHLOAD tool bar

4.2.1 Device Setup dialog

The device setup dialog allows to setup the target microcontroller. By selecting this menu item the following dialog window appears and offers the possibility to specify the corresponding V850 or RH850 based device file, the OCD security ID, the device clock (target frequency), the debug interface and the debug interface clock.

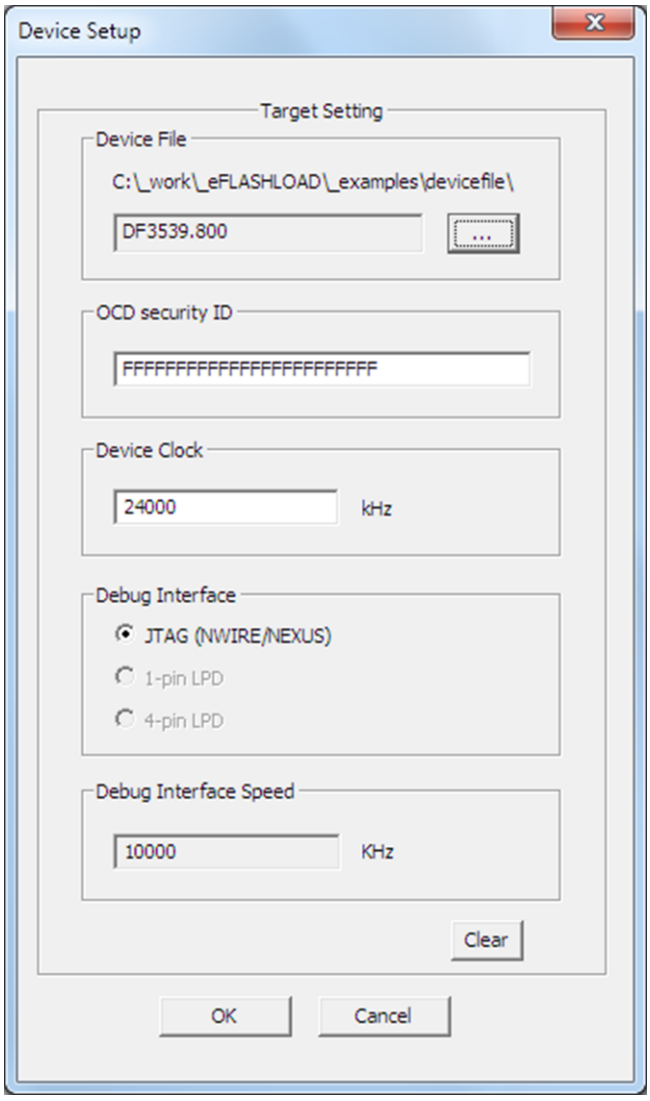


Figure 14 Device Setup Dialog

Notes:

- For V850 based products the Debug Interface and Debug Interface Clock can not be changed. The Debug Interface is fixed to JTAG (NWIRE/NEXUS) and the Debug Interface Speed is fixed to 10MHz.
- For RH850 based products only the 1-pin LPD or 4-pin LPD Debug Interface can be chosen.
    - For 1-pin LPD specify a debug interface speed (baudrate) of: **2000**, **1000** or **500** (Kbps).
    - For 4-pin LPD specify a debug interface speed (LPD clock) of: **16500**, **11000** or **5500** (KHz).

The <Clear> button offers the possibility to delete the current device settings. After the device setup has been finished the programmer parameter window is updated accordingly.

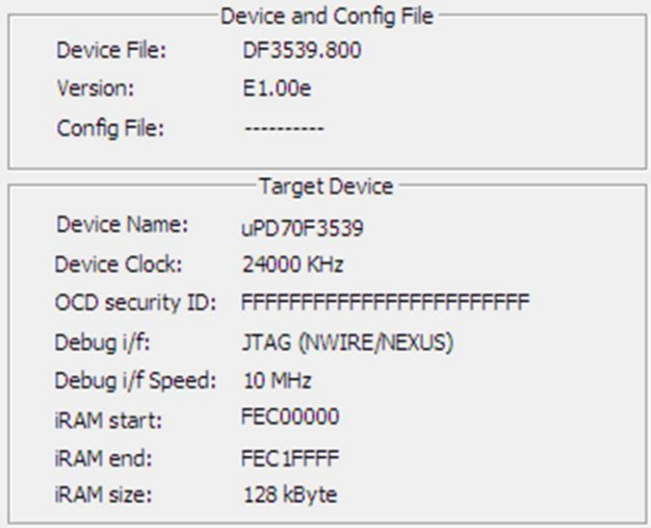


Figure 15 Device and Config file information

4.2.2 FLASH Monitor Setup dialog

The FLASH monitor setup dialog allows to select and setup the FLASH monitor program to be executed. By selecting this menu item the following dialog window appears and offers the possibility to specify the corresponding FLASH monitor, the FLASH monitor start address and the start address of the FLASH monitor parameter area.

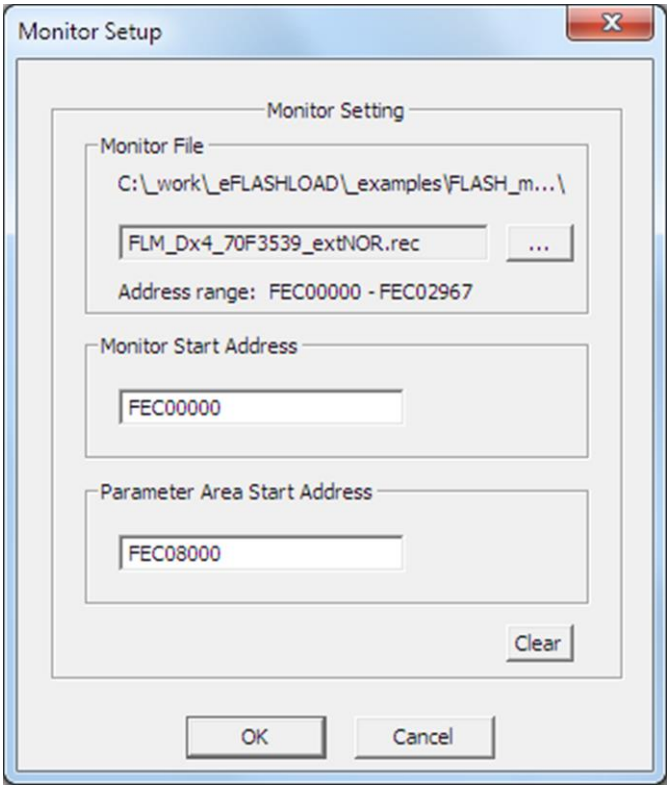


Figure 16 FLASH Monitor Setup dialog

The FLASH monitor file has to be available in an Intel-Hex or Motorola-S record format. For Intel-hex the extended linear address record and extended segment address record formats are supported. With regard to the Motorola-S record type the S3 record format is supported.

The FLASH monitor start address defines the entry address of the monitor program, equal to the reset address of the monitor program. Please note that the monitor start address must fit to the address range of the FLASH monitor file.

The FLASH parameter area start address defines the base address of the parameter and command area which is used for communication and synchronization between the eFLASHLOAD GUI and the FLASH monitor program. Please note that the parameter area start address must not point into the address area of the FLASH monitor file.

The <Clear> button offers the possibility to delete the current FLASH monitor settings. After the FLASH monitor setup has been finished the programmer parameter window is updated accordingly.

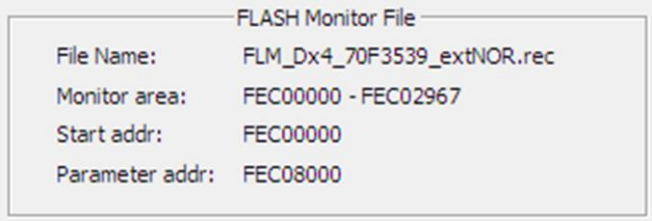


Figure 17 FLASH Monitor File information

4.2.3 FLASH Image setup dialog

The FLASH image setup dialog allows to select and setup the FLASH image to be programmed to the FLASH memory. By selecting this menu item the following dialog window appears and offers the possibility to specify the corresponding FLASH image and the address offset.

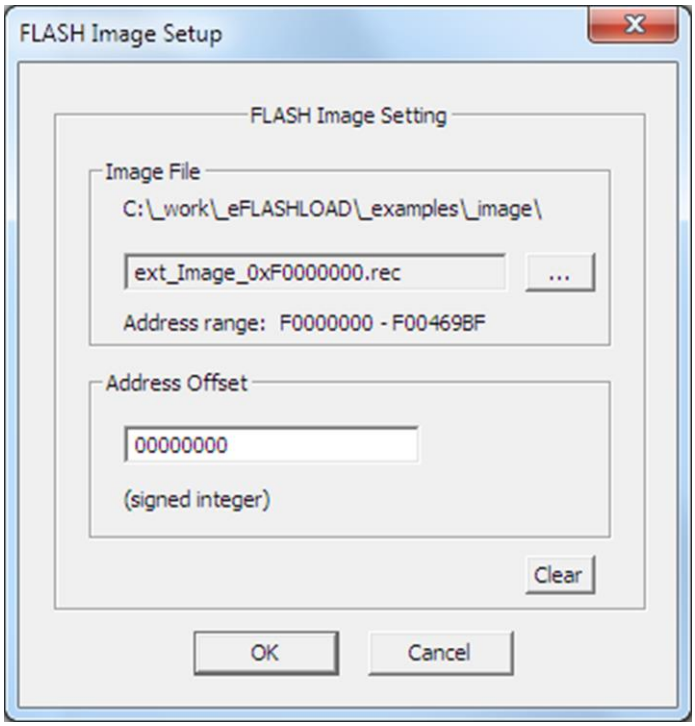


Figure 18 FLASH Image Setup dialog

The FLASH image file has to be available in an Intel-Hex or Motorola-S record format. For Intel-hex the extended linear address record and extended segment address record formats are supported. With regard to the Motorola-S record type the S2 and S3 record format is supported.

The FLASH image address offset can be specified as a signed integer. The resulting target address for the FLASH operation is displayed within the programmer parameter window after the FLASH image setup has been finished. The <Clear> button offers the possibility to delete the current FLASH image settings.

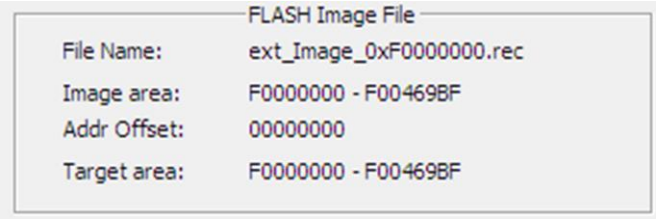


Figure 19 FLASH Image File information

4.2.4 Connect / Disconnect button

By pressing this button the debug interface connection to the target microcontroller is established. In case a connection was already established the communication is closed (disconnected). The button is only visible in case “Auto connect” is set to off.

4.2.5 Blank check button

By pressing this button the FLASH programmer GUI performs the blank check command.

4.2.6 Erase button

By pressing this button the FLASH programmer GUI performs the erase command.

4.2.7 Program button

By pressing this button the FLASH programmer GUI performs the program command.

4.2.8 Verify button

By pressing this button the FLASH programmer GUI performs the verify command.

4.2.9 E.P.V. button

By pressing this button the FLASH programmer GUI performs the sequential E.P.V. command (Erase -> Program -> Verify).

4.2.10 User button

By pressing this button the FLASH programmer GUI performs the user command.

4.2.11 Delete button

By pressing this button all configurations including the device, FLASH monitor and FLASH image settings were cleared.

4.3 Action log window

This window displays the log of programming GUI actions.

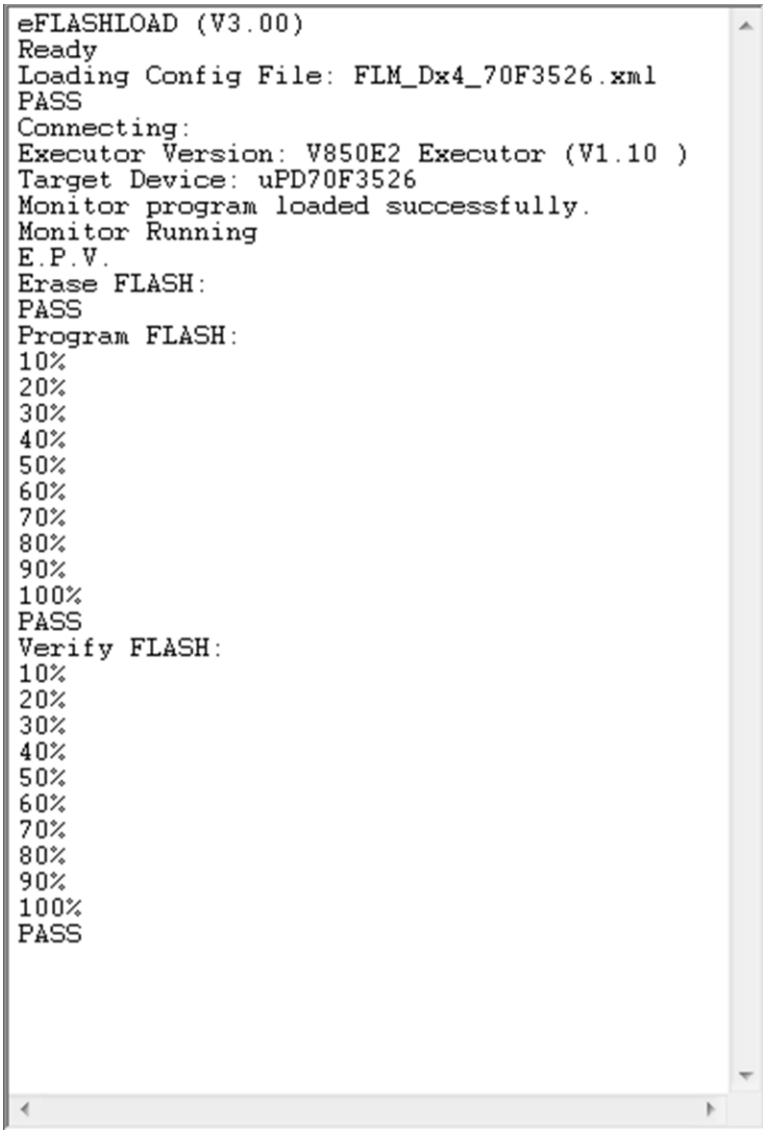


Figure 20 Action log window

4.4 Programming Parameter Window

This window displays the programming parameter settings.

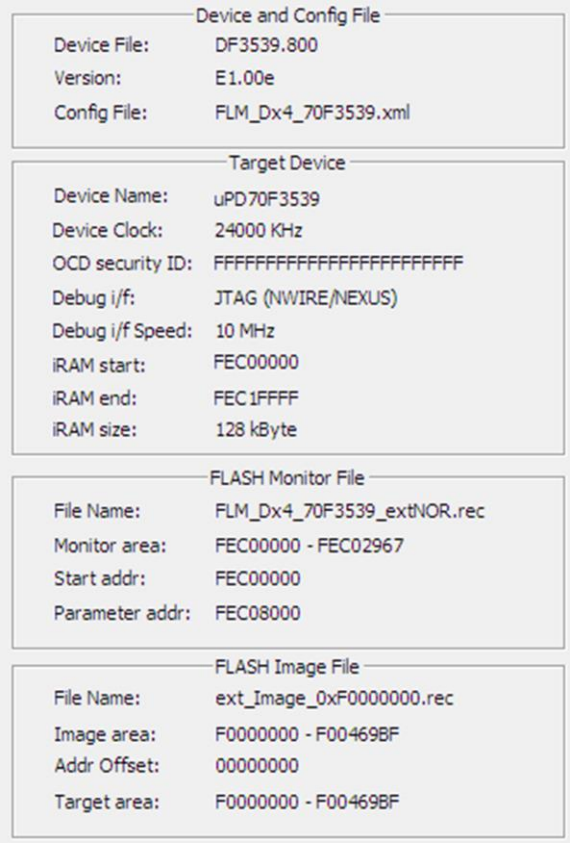


Figure 21 Programming Parameter Window

[Device and Config File] area

Displays information such as the device file, device file version and loaded configuration file.

[Target Device] area

Displays information such as the device name, OCD security ID code, device clock, debug interface, debug interface speed settings and target device iRAM mapping / resources.

[FLASH Monitor File] area

Displays information such as the FLASH monitor file, the monitor address area, start address and parameter area start address.

[FLASH Image File] area

Displays information such as the FLASH image file, the image address area, the address offset and target address area.

4.5 eFLASHLOAD Status bar

The status bar shows the current status and the progress of a command under execution.

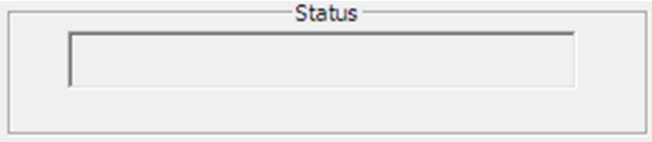


Figure 22 Status bar

	Immediately after the programming GUI is started
Connecting...	Establishing of the debug connection to the target device in progress.
Ready	The target device is ready to receive a new command
Running...	A command execution is in progress
Pass	A command execution is completed normally
Fail	A command execution is terminated abnormal

Table 10 Status bar

Chapter 5 Command line interface

Beside the graphical user interface the FLASH programmer GUI provides a command line interface to support batch mode operation or automatic FLASH operations. The syntax for the command line interface is as following:

eFLASHLOAD –xml <*config filename*> -<*command*>

By executing the embedded FLASH programmer GUI without any options from the command line the graphical user interface is started. Additionally, by starting the GUI only with the configuration file specified, the graphical user interface is started and configured according to the configuration file.

By running the embedded FLASH programmer GUI via the command line interface all status information, process information and operation results are displayed in the command shell window.

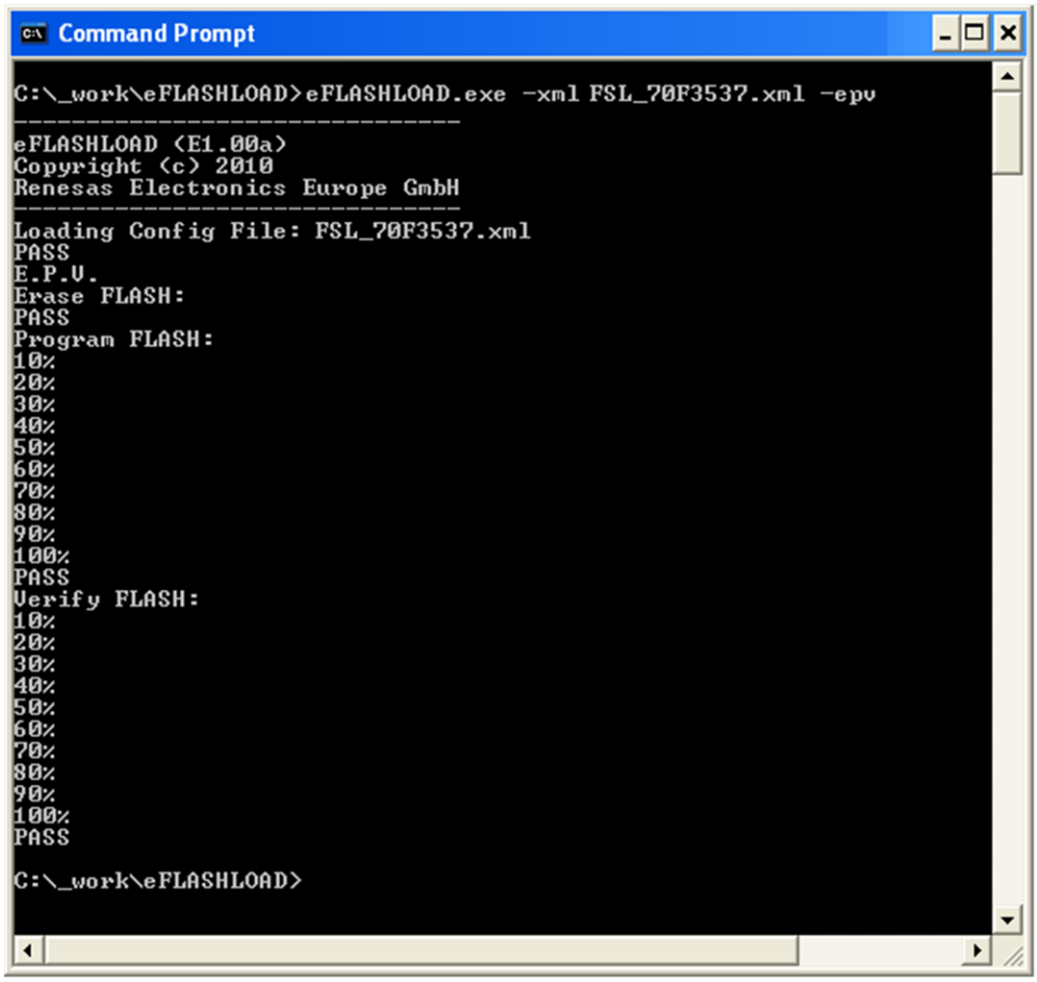


Figure 23 Command line interface

5.1 Command line options

The following command line options are supported by the embedded FLASH programmer:

Command	***Definition***
-xml <*configuration file*>	specifies the configuration file to be loaded
-bln	executes the [Blank Check] command
-ers	executes the [Erase] command
-prg	executes the [Program] command
-vrf	executes the [Verify] command
-epv	executes the [Autoprocedure(E.P.V.)] command. (Erase -> Program -> Verify).
-usr	executes the [User] command

Table 11 Command line options

By running the eFLASHLOAD programmer only one FLASH operation command can be passed at a time, example:

```
eFLASHLOAD -xml Dx4_NOR_FLASH.xml -epv
```

By passing more than one FLASH operation command to the eFLASHLOAD programmer an error message is displayed.

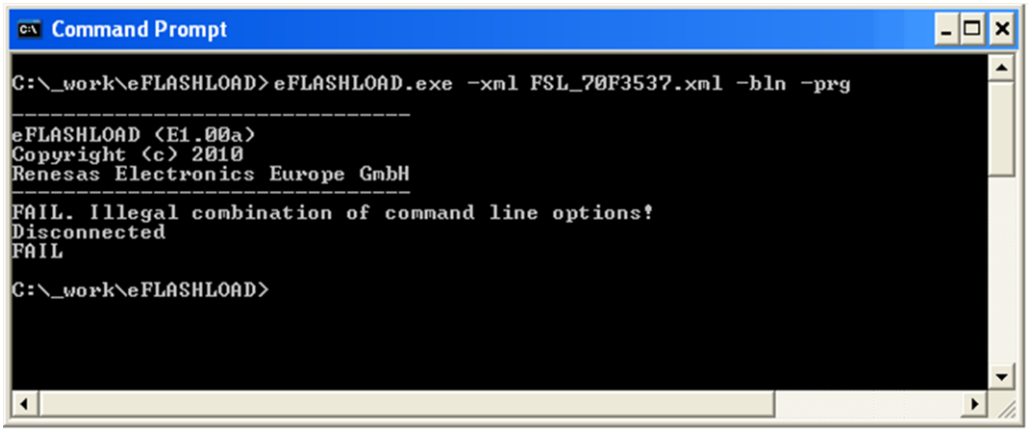


Figure 24 Command line error message

5.2 Return Codes

For using the embedded FLASH programmer within an automated programming flow, different return codes are generated depending on the FLASH operation result. In case the FLASH operation was successfully the FLASH programmer returns value “0” (zero). Otherwise in case of errors during FLASH operation the value “-1” (minus one) is returned by the embedded FLASH programmer software.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	January 07, 2011	-	First Edition Issued
2.00	March 15, 2011	-	Added V850ES and V850E1 Core support
2.01	September 24, 2012	-	Added E1 On-Chip Debug Emulator support
3.00	July 3, 2014	-	Added RH850 Core support Added User Command support
3.01	January 29, 2015	-	Aligned Bookmark Linkage

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2010 Renesas Electronics Corporation. All rights reserved.
Colophon 1.0

eFLASHLOAD

Renesas Electronics Corporation

R01UT0177ED0301